Multi-Objective Optimization for the Vehicle Routing Problem With Outsourcing and Profit Balancing

Zizhen Zhang, Hu Qin, and Yanzhi Li

***Abstract*— An importer in Hong Kong employs vehicles, all from external transport companies, to deliver products to its customers geographically scattered in different locations. The delivery plan needs to simultaneously minimize the total traveling cost and balance the profits among all transport companies. This transportation practice engenders a new variant of vehicle routing problems, called the vehicle routing problem with outsourcing and profit balancing (VRPOPB). The profits are balanced by maximizing the minimum unit profit of all transport companies, which can effectively avoid the occurrence of distorted solutions. We develop two multi-objective local search (MOLS) algorithms for the problem, where the second one enhances the first one by incorporating several additional techniques. To evaluate our algorithms, we conduct extensive experiments on 57 generated instances and a real case obtained from a food importer in Hong Kong. The computational results clearly demonstrate that our enhanced MOLS algorithm is able to achieve satisfactory solutions.**

***Index Terms*— Multi-objective optimization, vehicle routing problem, outsourcing, fairness, profit balancing.**

I. Introduction

VEHICLE routing is one of the most important fields in transportation systems. This paper studies an extension of vehicle routing problems in which the customer demand is fully outsourced to one of multiple transport companies. We call it the vehicle routing problem with outsourcing and profit balancing (VRPOPB). The background is from a large-scale importer in Hong Kong that supplies and distributes assorted products from its warehouse to hundreds of stores. Every day each store places a restock order with the importer to replenish its inventory. The logistics department of the importer outsources all delivery tasks to external transport companies through long-term contracts. Outsourcing is a better choice than keeping a large fleet of in-house vehicles for the importer that has both economic and management advantages. On one hand, since Hong Kong is one of the most competitive economics in the world, the transportation rates for common products have been fairly low, stable and predictable. Thus, the importer can concentrate more on its core business rather than being involved in peripheral logistics activities. On the other hand, outsourcing can help the importer convert fixed costs (e.g., labor costs, the cost of buying vehicles) into variable costs and avoid large expenditure on fixed assets.

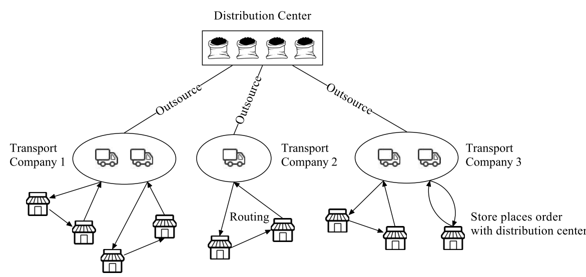


Fig. 1. An illustration of order outsourcing and vehicle scheduling.

There is an interesting phenomenon that more than 80% of the transport companies in Hong Kong are of small-scale and nearly half of them are self-employed. Each of these transport companies only owns a small number (e.g., two or three) of trucks and vans. According to the assigned orders, the transport companies dispatch their vehicles to the warehouse to collect products, and then deliver them to the corresponding stores. Since the stores are located in different geographically scattered points, each vehicle should be carefully scheduled to execute an efficient and economic delivery plan. Figure 1 illustrates a scheme of the order outsourcing and vehicle scheduling in the VRPOPB.

When only one transport company is considered, the problem becomes a *centralized* version. Although centralization is efficient with regard to business decisions, the importer will suffer from the negative effect of "putting all eggs in one basket". If the importer uses multiple transport companies, i.e., *decentralized delivery strategy*, it can gain benefit from the competition of the transport companies and can easily find an alternative when some transport company is unable to provide satisfied service. However, decentralization may bring fairness issue among multiple transport companies.

Manuscript received January 5, 2017; revised July 11, 2017, January 1, 2018, July 12, 2018 and October 31, 2018; accepted March 31, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 71390335, Grant 71571077, Grant 71601191, and Grant 71531009, and in part by the Guangdong Natural Science Funds under Grant 2016A030313264. The Associate Editor for this paper was C. C. White. *(Corresponding author: Yanzhi Li.)*

Z. Zhang is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China, and also with the Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510275, China.

H. Qin is with the School of Management, Huazhong University of Science and Technology, Wuhan 430074, China.

Y. Li is with the Shenzhen Research Institute and the College of Business, City University of Hong Kong, Hong Kong (e-mail: yanzhili@cityu.edu.hk).

Digital Object Identifier 10.1109/TITS.2019.2910274

Many vehicle routing problems (VRPs) in existing literature only consider centralized delivery strategy and focus on minimizing the total traveling cost, which accounts not just for the travel distance, but also for the amount of greenhouse emissions, fuel and traveling times. In the VRPOPB, besides *traveling cost minimization*, we also need to consider the fairness issue by making the profits of every transport companies as close as possible, which is called *profit balancing*. The profit of a transport company is the difference of the rewards obtained by fulfilling the assigned orders and the total transportation cost. In practice, two transport companies with similar conditions (e.g., scale, qualification, etc.) are supposed to gain comparable profits. Otherwise, the transport company with less profit will complain and tend to raise the quotation in later cooperation, thereby impairing the benefit of the importer in the long run.

Based on the above discussions, we can identify two objectives in the VRPOPB, namely traveling cost minimization and profit balancing. Minimizing traveling cost is important not only to reduce transport companies' cost and also to reduce pollution, especially to metropolises like Hong Kong. These two objectives are equally important but sometimes in conflict with each other. For example, balancing the profit may require the reassignment of orders to transport companies and lead to vehicle routes with larger traveling costs. Hence, the VRPOPB is essentially a bi-objective optimization problem. A general approach for the multi-objective optimization problem is to generate a representative subset of *Pareto-optimal* solutions. A solution is called Pareto-optimal if it is impossible to make any one objective value better off without making at least one objective value worse off [1]. The decision maker can make trade-off between different objectives and select the most appropriate Pareto-optimal solution based on its experience or preference.

The goal of this study is to help the importer automatically assign orders to different transport companies and design routes for vehicles such that the total traveling cost is minimized and the profits of all transport companies are balanced. It is worth noting that the scheduling of vehicles in real practice is very complicated. The solution of the VRPOPB actually acts as a simplified plan under certain limitations. The transport companies are allowed to make appropriate adjustments on the vehicle routes in the real operations.

In addition to making methodological contributions, this paper primarily intends to present a practically motivated problem and design a comprehensive, customized approach to the problem by deliberately combining a variety of techniques from the literature. Extensive numerical studies on both existing and generated test instances demonstrate that the proposed approach is very effective.

The remainder of the paper is structured as follows. Section II gives an overview of the relevant research in existing literature. In Section III, we present a formal definition, formulation and some properties of the VRPOPB. In Section IV, two multi-objective optimization algorithms are designed for the problem. In Sections V and VI, we conduct a series of experiments to prove the performance of our algorithms and provide a case study. Finally, in Section VII we conclude our article with some closing remarks and suggest some possible directions for future research.

II. Related Work

Outsourcing has been implicitly considered in many VRPs. For example, in the team orienteering problems [2], [3] and the VRP with profits [4]–[6], a fixed number of vehicles are dispatched to collect as many profits from the customers as possible and the unserved customers are assumed to be outsourced to external service providers. However, the previous articles that highlight "outsourcing" in VRPs are scarce. Moon *et al.* [7] incorporate overtime and outsourcing in the VRP with time windows (VRPTW), where a third party logistics company has a limited number of vehicles, the drivers are allowed to work overtime, and customer demands may be outsourced to external vehicles. In this problem, the decision maker has to consider the tradeoff between the overtime and outsourcing costs. They developed a mixed integer programming model, a genetic algorithm and a hybrid algorithm for their problem. Lee *et al.* [8] studied a container transportation problem for a logistics company in Singapore who cannot handle all jobs due to the insufficient number of vehicles and has to outsource some jobs to other transport companies. They built a vehicle capacity planning system, which includes a tabu search procedure, to generate high-quality solutions for the problem. Zäpfel and Bögl [9] investigated a very complex vehicle routing and crew scheduling problem that takes into consideration variable vehicle capacities, time windows, multi-period and personnel planning, and allows some tours and drivers to be outsourced to subcontractors for lower total cost. Li *et al.* [10] investigated a transportation service subcontracting problem, which is based on an outsourcing practice in a large limousine fleet company. When the company has demand exceeding its capacity, it subcontracts some of its currently booked rides to other smaller affiliated fleets.

The literature related to balancing workloads or profits in VRPs is briefed as follows. Lee and Ueng [11] introduced the VRP with load-balancing that aims to minimize the total traveling distance and balance the workloads among employees as much as possible. The balancing objective requires that the sum of the working time difference between each vehicle and the vehicle with the smallest working time is as small as possible. Nikolakopoulou *et al.* [12] tried to balance the time utilization of the vehicles used in distribution networks. They devised a heuristic to minimize the gap between the maximum and minimum completion times among all vehicles. Jozefowiez *et al.* [13] modeled the VRP with route balancing into a bi-objective optimization problem, where the first objective is to minimize the total length of all routes and the second one is to minimize the difference between the maximum and minimum route lengths. Kritikos and Ioannou [14] studied a VRPTW variant that takes into account balancing the loads carried by active members of the vehicle fleet. The objective function of this problem is the weighted sum of route costs, vehicle costs and the imbalance of the vehicle loading. They proposed a new approach that is based on the free disposal hull method of data envelopment analysis to produce very promising solutions for the problem.

Huang *et al.* [15] studied a humanitarian relief routing problem, which considers the decisions on vehicle routing and supply allocation. Other than minimizing cost, the objectives also include efficacy and equity. The equity issue is measured by deviations between recipients in efficacy. They analyzed the structure of vehicle routes and the distribution of resources and proposed routing principles for humanitarian relief. Pacheco *et al.* [16] conducted research on bus routing problem and formulated it as a bi-objective optimization model. The goal is to minimize both the longest route length and the total route length. Tabu search within a Multiobjective Adaptive Memory Programming framework was proposed to solve the problem. Motivated by a real world problem in Tenerife, Spain, Melián-Batista *et al.* [17] addressed a bi-objective VRPTW which minimizes the total traveling costs and the difference between the length of the longest route and the shortest one. They developed a scatter search algorithm to identify the Pareto-optimal solutions of the problem.

As can be seen from the aforementioned literature, the multi-objective VRPs have been receiving more and more attention in recent years. The cost and fairness are usually two objectives that need to be considered simultaneously. Minimizing cost is important to both reduce vehicles' transportation cost and to reduce vehicles emissions. Fairness is critical to ensures satisfaction among drivers, employees, and/or 3PLs (as in our case). For more knowledge on the multi-objective VRPs, we refer the reader to Jozefowiez *et al.* [18]–[20]. There are two main features that distinguish our VRPOPB from other VRPs in the literature. First, the orders are completely outsourced and the transport company attains profit by fulfilling the assigned orders. Second, the fairness objective requires balancing the profits among transport companies rather than individual vehicles. To the best of our knowledge, there is no existing literature addressing the VRPOPB.

III. Problem Definition, Formulation and Properties

The VRPOPB is defined on a complete and undirected graph $G = (N, E)$, where $N = \{0, 1, \ldots, n\}$ is the node set and $E = \{(i, j) \mid i, j \in N\}$ is the edge set. The set N consists of depot 0 and n stores (or called *customers*). The transportation cost on each edge $(i, j) \in E$ is denoted by c_{ij}. Each customer $i \in N \setminus \{0\}$ corresponds to an order that has a demand d_i and a reward w_i. The demand of an order cannot be split and the reward is collected once the order is fulfilled. For notational convenience, we assume that $d_0 = 0$ and $w_0 = 0$. A collection of m vehicles (denoted by set V) with capacities $C_1, \ldots, C_m$ from all transport companies are initiated at the depot and to be dispatched to fulfill orders. The vehicles must return to the depot after completing all assigned orders. Let T be the set of transport companies and $V(t)$ be the set of vehicles owned by the transport company $t \in T$. Since each vehicle is managed by exactly one transport company, we have $V(t_1) \cap V(t_2) = \emptyset$ if $t_1 \neq t_2$ and $V = \bigcup_{t \in T} V(t)$.

Define a binary decision variable y_{ik} that is equal to 1 if customer i is served by vehicle k, and 0 otherwise. Let a binary decision variable x_{ijk} be the number of times that vehicle k traverses edge (i, j) and a continuous variable u_{ik} be the remaining capacity of vehicle k before visiting customer i. We introduce a bi-objective optimization model for the VRPOPB by extending the arc-flow model of the capacitated VRP (CVRP) as follows:

$$\text{(Model } M_1) \ \Big(opt\ f_1(\mathbf{x}, \mathbf{y}),\ opt\ f_2(\mathbf{x}, \mathbf{y}) \Big) \tag{1}$$

$$\text{s.t.} \sum_{k \in V} y_{ik} = 1, \quad \forall\, i \in N \setminus \{0\} \tag{2}$$

$$\sum_{k \in V} y_{0k} = m \tag{3}$$

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{jik} = y_{ik}, \quad \forall\, i \in N, k \in V \tag{4}$$

$$\sum_{i \in N} d_i y_{ik} \le C_k, \quad \forall\, k \in V \tag{5}$$

$$u_{ik} - u_{jk} + C_k x_{ijk} \le C_k - d_j, \quad \forall\, i, j \in N \setminus \{0\}, i \neq j, k \in V, d_i + d_j \le C_k \tag{6}$$

$$d_i \le u_{ik} \le C_k, \quad \forall\, i \in N \setminus \{0\}, k \in V \tag{7}$$

$$y_{ik} \in \{0, 1\}, \quad \forall\, i \in N, k \in V \tag{8}$$

$$x_{ijk} \in \{0, 1\}, \quad \forall\, i, j \in N, k \in V \tag{9}$$

The sign *opt* can be either *max* or *min*. Functions $f_1(\mathbf{x}, \mathbf{y})$ and $f_2(\mathbf{x}, \mathbf{y})$ are associated with traveling cost minimization and profit balancing, respectively. Constraints (2) state that each customer must be served by exactly one vehicle. Constraints (3) guarantee that all vehicles must start from the depot. Constraints (4) are flow conservation constraints. Constraints (5) ensure that vehicle capacity must be respected. Constraints (6) and (7) are subtour elimination constraints, which were originally proposed for the traveling salesman problem (TSP) by Miller *et al.* [21]. The first objective function can be written as:

$$f_1(\mathbf{x}, \mathbf{y}) = \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \tag{10}$$

To present the second objective function, we denote by p_k the profit gained by vehicle k. As shown in Expression (11), p_k is the difference between the collected rewards and the traveling cost of vehicle k.

$$p_k = \sum_{i \in N} w_i y_{ik} - \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \tag{11}$$

Since different transport companies may have different scales and maintain different numbers of vehicles, it is more rational to balance the unit profit than the total profit. Let W_t be the scale of transport company t and $\bar{p}_t$ be the unit profit of transport company $t \in T$. We have

$$\bar{p}_t = \frac{1}{W_t} \sum_{k \in V(t)} p_k \tag{12}$$

A commonly used indicator for profit balancing is the difference between the maximum and minimum unit profits of transport companies, namely, $\max_{t \in T} \bar{p}_t - \min_{t \in T} \bar{p}_t$. However, minimizing this indicator may result in distorted solutions, which is illustrated in Figure 2. A solution s is called *distorted*, if s is Pareto-optimal and some route(s) is not optimal in terms of the traveling cost. Figure 2 shows two vehicle

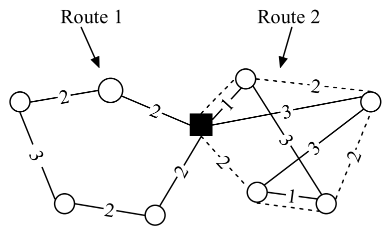


Fig. 2. An example of the distorted solution.

routes marked in solid lines, where the square and circles represent the depot and customers, respectively. Suppose that the reward of each customer is 10 and there are two transport companies with the same scale. Routes 1 and 2 have the same traveling cost and profit, which are 11 and 29, respectively. Thus, the difference between the maximum and minimum unit profits is 0. The objective values of this solution can be represented by (22, 0). If we replace Route 2 with a route indicated by the dash lines, the traveling cost and profit of this vehicle become 8 and 32, respectively. Then the objective values of the resultant solution is $(19, 3)$. Although both of these two solutions are Pareto-optimal, the previous one is distorted because Route 2 takes a detour with a larger traveling cost. Obviously, if we take this indicator, the Pareto-optimal solutions found by our approaches may include many distorted solutions, which will make the decision process inefficient.

It is noteworthy that Jozefowiez *et al.* [13] have studied the VRP with route balancing and mentioned the artificially balanced solutions, which essentially are distorted solutions. They used a TSP improvement procedure to optimize the route length. However, their approaches cannot fundamentally eliminate the distorted solutions as the TSP itself is an NP-complete problem.

To prevent distorted solutions, we introduce the *max_min* fairness, which defines $f_2(\mathbf{x}, \mathbf{y})$ as:

$$\begin{aligned} f_2(\mathbf{x}, \mathbf{y}) &= \min_{t \in T} \bar{p}_t \\ &= \min_{t \in T} \frac{1}{W_t} \sum_{k \in V(t)} (\sum_{i \in N} w_i y_{ik} - \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}) \end{aligned} \quad (13)$$

The profit balancing is achieved by maximizing $f_2(\mathbf{x}, \mathbf{y})$. According to Theorem 1, the *max_min* fairness will not distort the Pareto-optimal solutions. For example, in Figure 2, whether Route 2 is marked in the solid lines or the dash lines, the corresponding solutions have the same value of $f_2(\mathbf{x}, \mathbf{y})$, which is equal to 29. Therefore, $(19, 29)$ is the only Pareto-optimal solution of the example in Figure 2.

Theorem 1: For a Pareto-optimal solution $s = (\mathbf{x}, \mathbf{y})$ with two objectives $f_1(s)$ and $f_2(s)$ defined by Expressions (10) and (13), each route of s must be an optimal TSP tour.

Proof: Proof by contradiction. Assume that s is Pareto-optimal with some route r which is not TSP optimal. Then, route r can be improved into a TSP optimal route r', and therefore a solution s' can be obtained by replacing r with r' in solution s.

We have $f_1(s') < f_1(s)$, since r' is better than r in terms of traveling cost. We have $f_2(s') \geq f_2(s)$, because the collected rewards by each transport company do not change and the traveling cost of route r reduces. Therefore, s is dominated by s', which contradicts with the assumption that s is Pareto-optimal. □

Therefore, the model of the VRPOPB can be written as:

$$\Big(\min f_1(\mathbf{x}, \mathbf{y}), \;\; \max f_2(\mathbf{x}, \mathbf{y})\Big) \quad (14)$$

or

$$\min \;\Big(f_1(\mathbf{x}, \mathbf{y}), \;\; -f_2(\mathbf{x}, \mathbf{y})\Big) \quad (15)$$

subject to Constraints (2) – (9). We can integrate these two objectives $f_1(\mathbf{x}, \mathbf{y})$ and $f_2(\mathbf{x}, \mathbf{y})$ into a single objective by introducing weights α_1 and α_2. When $\alpha_1 = 1$ and $\alpha_2 = 0$, the VRPOPB becomes the traditional CVRP. When $\alpha_1 = 0$, $\alpha_2 = 1$ and $|V(t)| = 1$ for every $t \in T$, the VRPOPB reduces to the bottleneck generalized assignment problem [22]. Both of these two special cases are NP-hard and intractable.

IV. Solution Approaches

We develop multi-objective evolutionary algorithms (MOEAs) to handle the problem. By evolving a population of solutions, MOEAs are able to find an approximate Pareto-optimal set [23]. Popular MOEAs include NSGA-II (Non-dominated Sorting Genetic Algorithm II) [24] and MOEA/D (Multiobjective Evolutionary Algorithm based on Decomposition) [25]. The MOEAs have been successfully applied to various multi-objective optimization problems; examples can be found in Tan *et al.* [26], Liu *et al.* [27]. Since the VRPOPB is a discrete optimization problem, we employ problem-specific heuristics, such as local search refinements, as the main ingredients in our MOEAs. In this section, we first introduce some preliminaries and then devise a standard multi-objective local search (MOLS) algorithm Tricoire [28], which is one type of MOEAs. Finally, we enhance MOLS to MOLS+ by adding several useful techniques.

A. Preliminaries

Let S be the set of feasible solutions of the VRPOPB and $O = \{o \in R^2 : o = (f_1(s), f_2(s)), \; \forall \; s \in S\}$ be the set of number pairs, where the functions $f_1(s)$ and $f_2(s)$ are given in Equation (10) and (13), respectively. Some basic concepts are given as follows.

Definition 1: A feasible solution $s_1 \in S$ is said to strictly *dominate* another feasible solution $s_2 \in S$ (denoted by $s_1 \prec s_2$), if and only if both of the following conditions hold:

1) $f_1(s_1) \leq f_1(s_2)$ and $-f_2(s_1) \leq -f_2(s_2)$,
2) $f_1(s_1) < f_1(s_2)$ or $-f_2(s_1) < -f_2(s_2)$.

Definition 2: Given a set of feasible solutions P, the *non-dominated set* P' contains the solutions that are not dominated by any member in P. The non-dominated set of the feasible search space S is called the *Pareto-optimal set*. The image of the Pareto-optimal solutions in the objective space is called the *Pareto front*.

Note that the solution space of the VRPOPB is the same as that of the capacitated VRP. Thus, finding all Pareto-optimal solutions is also very difficult and even may be impossible due to the intractability of the problem. Alternatively, it is more

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 The Multi-Objective Local Search Algorithm

1: Generate an initial population P;
2: **while** the number of generations does not exceed N_{gen} **do**
3: $\quad P' \leftarrow \emptyset$;
4: $\quad$ **for** $l \leftarrow 1$ to $Iter$ **do**
5: $\quad\quad$ Randomly generate values for weights $\omega = (\omega_1, \omega_2)$;
6: $\quad\quad s \leftarrow$ **select_a_solution**(P, ω);
7: $\quad\quad$ **for** $k \leftarrow 1$ to K **do**
8: $\quad\quad\quad P' \leftarrow P' \cup \{\mathbf{LS}_k(s)\}$;
9: $\quad\quad$ **end for**
10: $\quad$ **end for**
11: $\quad P \leftarrow$ **update**(P, P');
12: **end while**
13: **return** P.

reasonable to find a set of non-dominated solutions that are as close to the Pareto-optimal set as possible and as diverse as possible [29].

B. Multi-Objective Local Search Algorithm

The MOLS algorithm employs different local search operators in the multi-objective evolutionary process. Each operator is applied to improve a solution in a certain direction. A population is used to preserve a set of promising solutions. Algorithm 1 presents the framework of MOLS. The block within the inner loop can be executed *Iter* times in each outer iteration. At the beginning of the inner loop, we randomly generate a weight vector $\omega = (\omega_1, \omega_2)$, where $\omega_1 + \omega_2 = 1$, and then select a solution from P based on ω by invoking function **select_a_solution**(P, ω) (see lines 5 – 6, Algorithm 1). We generate K Pareto-optimal solutions by applying each of K local search operators to the selected solution and then add them in P' (see lines 7 – 8, Algorithm 1). The function $\mathbf{LS}_k(s)$ realizes the local search procedure with regard to the k-th local search operator and outputs the resultant local optimal solution. The algorithm terminates when the number of generations exceeds N_{gen}. As the solution structure of the VRPOPB is the same as that of the CVRP, most of the local search operators for the CVRP can be employed. In the following context of this subsection, we will detail the process of generating the initial population, the local search operators, the selection strategy in function **select_a_solution**(P, ω) and the population updating strategy in function **update**(P, P').

1) Initial Population: The initial population is composed of a set of solutions, which are constructed using a two-stage procedure. The first stage is inspired by the *cluster-first-route-second* method [30], which divides customers into groups using the *sweep algorithm* [31], [32]. The sweep algorithm works as follows. Assume that the depot is the pole (i.e., the origin of a Cartesian system) and all customers have Cartesian coordinates corresponding to their locations. We use (θ_i, ρ_i) to represent the polar coordinate of customer i, where θ_i is the angle and ρ_i is the ray length. All customers are sorted in ascending order of θ_i, and then in ascending order of ρ_i if ties are encountered. Suppose that n customers are rearranged to form a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, where π_i is the index of a customer. We invoke a greedy procedure to divide π into m groups. The greedy procedure first employs an empty vehicle. Next, it sequentially selects an unserved customer from π and tries to assign it to one of the used vehicles. If no used vehicle has enough capacity to serve the customer, a new vehicle is employed. After assigning all customers to the vehicles, we optimize each vehicle route using a TSP solver.

After the first stage, we are able to identify the profit p_k of each vehicle k (i.e., the collected rewards minus the traveling cost). The second phase is to balance the unit profit among transport companies using an integer programming model. To construct this model, let z_{tk} be the binary decision variable indicating whether vehicle route k is assigned to transport company t, and $\bar{p}_{min}$ be the decision variable representing the minimum unit profit. The following model is able to balance the unit profits among all transport companies.

$$\text{(Model } M_2\text{)} \quad \max \ \bar{p}_{min} \tag{16}$$

$$\text{s.t.} \ \sum_{k \in V} z_{tk} = |V_t|, \quad \forall\ t \in T \tag{17}$$

$$\sum_{t \in T} z_{tk} = 1, \quad \forall\ k \in V \tag{18}$$

$$\bar{p}_{min} \leq \frac{1}{W_t} \sum_{k \in V} p_k z_{tk}, \quad \forall\ t \in T \tag{19}$$

$$z_{tk} \in \{0, 1\}, \quad \forall\ t \in T, k \in V \tag{20}$$

$$\bar{p}_{min} \geq 0 \tag{21}$$

The objective (16) is to maximize the minimum unit profit $\bar{p}_{min}$. Constraints (17) and (18) ensure that each transport company t possesses exactly $|V_t|$ vehicles and each vehicle route must be assigned to exactly one transport company. Constraints (19) guarantee that the unit profit of transport company t must be greater than or equal to $\bar{p}_{min}$. The problem (16)–(21) can be proved to be NP-complete by reducing it to the so-called Linear Partition Problem: given a set of non-negative numbers and an integer k, partition the set into k ranges so as to minimize the maximum sum over all the ranges. However, in practice the cardinalities of T and V are relatively small, so the model can be easily solved to optimality by general IP solvers (e.g., ILOG CPLEX).

One execution of the above two-stage procedure generates only one feasible VRPOPB solution. To obtain more feasible solutions, we can rotate the polar coordinate system and then get different customer permutations. This is equivalent to setting $\pi = (\pi_{(1+i)\ mod\ n+1}, \ldots, \pi_{(n+i)\ mod\ n+1})$, where i is a randomly generated integer from 0 to $n-1$. The remaining steps are the same as the aforementioned.

2) Local Search Operators: The neighborhoods of a given solution are defined by the following four local search operators (i.e., $K = 4$), which are widely used in the approaches for many existing VRPs. These neighborhoods only consider the feasible solutions. After each operation, the objectives of the new solution will be updated.

1) *exchange*(i, j). This operator exchanges the positions of customers i and j to generate a new solution.

2) *relocate*(i, j). This operator removes customer i from its current position and then inserts it to the position before node j (node j can be either depot or customer node).
3) *2-opt*(i, j). This operator is applied to two different vehicle routes. Suppose that the two routes associated with customers i and j are $(0, \ldots, i, i+1, \ldots, 0)$ and $(0, \ldots, j, j+1, \ldots, 0)$, respectively. The operator replaces the edges $(i, i+1)$ and $(j, j+1)$ with $(i, j+1)$ and $(j, i+1)$, generating two new routes $(0, \ldots, i, j+1, \ldots, 0)$ and $(0, \ldots, j, i+1, \ldots, 0)$.
4) *reverse*(i, j). This operator is applied to one vehicle route and reverses the segment (i, j) of the route $(0, \ldots, i, i+1, \ldots, j-1, j, \ldots, 0)$ to make a new route $(0, \ldots, j, j-1, \ldots, i+1, i, \ldots, 0)$.

The size of each local search neighborhood is $O(n^2)$, which is a relatively small number. We denote by s' the resultant solution after applying some operator to s. The local search operator in MOLS tries to improve the solution according to the following guiding function:

$$f(s, \omega) = \omega_1 \frac{f_1(s) - f_1^{min}}{f_1^{max} - f_1^{min}} - \omega_2 \frac{f_2(s) - f_2^{min}}{f_2^{max} - f_2^{min}}, \qquad (22)$$

where f_k^{min} and f_k^{max} $(k = 1, 2)$ are the minimum and maximum values for the k-th objective of the solutions in the population. The second term on the right-hand-side of the equation has a negative sign because $f_2(s)$ is to be maximized. With this guiding function, we accept s' if $f(s', \omega) < f(s, \omega)$.

3) Solution Selection and Population Updating: A solution s is selected from the population P using the tournament selection strategy [33] (see line 6, Algorithm 1), which involves running several "tournaments" among individuals chosen at random from the population and selects the winner. Specifically, it first chooses $\tau \cdot |P|$ solutions from P at random, where $\tau \in (0, 1)$ is a user-defined parameter. The solution with the smallest value of $f(s, \omega)$ is finally chosen from the tournament.

The new population P is updated by producing a set of solutions from $P \cup P'$ (see line 11, Algorithm 1). We invoke the *non-dominated sorting* approach and the *crowded-comparison* approach, which were proposed in NSGA-II, to preserve a set of high-quality and diverse solutions. Readers are referred to Deb *et al.* [24] for more details of the updating procedure.

C. Enhancement of MOLS

The MOLS algorithm is easy to be implemented, but it does not fully utilize the characteristics of the VRPOPB. In order to further improve the solution quality, we enhance MOLS by introducing the giant-tour representation, the recombination operator and the large neighborhood search procedure. Algorithm 2 shows the enhanced multi-objective local search algorithm (called MOLS+ for short). This algorithm is motivated by the memetic algorithm [34], which is a population-based approach with separate individual local improvement procedures. Function **recombination**(P) produces a new population P' using a crossover operator, which combines the information of two parents to generate new offsprings. A large

Algorithm 2 The Enhanced Multi-Objective Local Search Algorithm

Generate a set P of non-dominated solutions;
while the number of generations does not exceed N_{gen} **do**
  $P' \leftarrow$**recombination**(P);
  $P'' \leftarrow \emptyset$;
  for $l \leftarrow 1$ to $Iter$ **do**
    $s \leftarrow$ **select_a_solution**$(P \cup P')$;
    $s \leftarrow$ **LNS**(s);
    for $k \leftarrow 1$ to K **do**
      $P'' \leftarrow P'' \cup \{\mathbf{LS}_k(s)\}$;
    end for
  end for
  $P \leftarrow$ **update**(P, P'');
end while
return P;

neighborhood search procedure is implemented in function **LNS**(s) to further improve the solution s.

1) Giant-Tour Representation: The giant-tour representation [35] is a compact encoding mode for the VRPs, and can be converted into an optimal VRP solution (subject to the visiting order) using a *splitting* procedure. The design of the recombination process becomes easy with the help of the giant-tour representation.

Given a solution consisting of m vehicle routes, suppose that the j-th route is $r_j = (r_j^1, r_j^2, \ldots, r_j^{R(j)})$, where $R(j)$ is the number of customers visited and r_j^i is the index of the customer in the i-th position of the route. Note that we omit the depot in this route representation. Denote by "$\oplus$" the concatenation operator that connects the last customer in one route and the first customer in the other route, namely, $r_i \oplus r_j = (r_i^1, \ldots, r_i^{R(i)}, r_j^1, \ldots, r_j^{R(j)})$. The giant tour is the concatenation of all vehicle routes as shown by Expression (23). Since each customer can be visited exactly once, the giant tour is essentially a permutation of customers.

$$r = r_1 \oplus r_2 \oplus \ldots \oplus r_m \qquad (23)$$

With the giant-tour representation, the solution space of the VRPOPB includes all customer permutations. On the other hand, given a giant tour r, we need to find a way to split r into multiple feasible vehicle routes. As the VRPOPB has two objectives, two splitting methods, which are actually dynamic programming algorithms, are proposed in Sections IV-C2 and IV-C3.

2) Splitting Method for Traveling Cost Minimization: Given a giant tour r represented by a permutation $(\pi_1, \pi_2, \ldots, \pi_n)$, our goal is to split r into m consecutive segments, where each segment corresponds to a feasible vehicle route. The first splitting method aims at finding a splitting that minimizes the total route length. We denote by a state $L[k][i]$ the minimum total traveling cost obtained by using k vehicles to visit $(\pi_1, \pi_2, \ldots, \pi_i)$. The dynamic programming recursion is:

$$L[k][i] = \min_{0 \le j \le i \text{ and } cap(j+1,i) \le C} \Big\{ L[k-1][j] + cost(j+1, i) \Big\}, \qquad (24)$$

where $cost(j,i)$ and $cap(j,i)$ correspond to the traveling cost and the required capacity of route $(0, \pi_j, \ldots, \pi_i, 0)$, respectively (see Expressions (25) and (26)). Expression (24) states that the value of $L[k][i]$ depends on each $L[k-1][j]$ and the traveling cost of vehicle k visiting $(0, \pi_{j+1}, \ldots, \pi_i, 0)$.

$$cost(j,i) = \begin{cases} c_{0,\pi_j} + c_{\pi_i,0} + \sum_{j'=j}^{i-1} c_{\pi_{j'},\pi_{j'+1}}, & \text{if } j \leq i; \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

$$cap(j,i) = \begin{cases} \sum_{j'=j}^{i} d_{\pi_{j'}}, & \text{if } j \leq i; \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

The boundary condition of the dynamic programming is $L[0][0] = 0$ and we need to obtain the value of $L[m][n]$, i.e., the minimum total traveling cost achieved by dispatching m vehicles to serve all customers. This dynamic programming that has a time complexity of $O(mn^2)$ is essentially the *Bellman's* method for the shortest path problem with at most m arcs. We refer the reader to Prins *et al.* [36] for more details.

It is worth noting that a certain giant tour may be infeasible, i.e., we cannot find a valid splitting such that each of the associated routes satisfies the capacity constraint. In such situation, we try to find the biggest n' that satisfies $n' < n$ and calculate the value of $L[m][n']$. The remaining customers $\{\pi_{n'+1}, \ldots, \pi_n\}$ are unserved and thus penalty costs can be added.

3) Splitting Method for Profit Balancing: Directly finding a splitting to maximize the minimum unit profit is difficult, as different transport companies are very likely to have different numbers of vehicles. We instead seek a splitting that makes the overall minimum route profit (namely, $\min_{k=1}^{m} p_k$, MRP for short) maximal.

Let $F[k][i]$ be the maximum value of the MRP obtained by using k vehicles to serve $(\pi_1, \ldots, \pi_i)$ in order. The dynamic programming recursion is given as:

$$F[k][i] = \max_{0 \leq j \leq i \text{ and } cap(j+1,i) \leq C} \Big\{ \min\{F[k-1][j], pro(j+1,i)\} \Big\}, \tag{27}$$

The term $pro(j,i)$ is the profit of the route $(0, \pi_j, \ldots, \pi_i, 0)$ that is calculated by:

$$pro(j,i) = \begin{cases} \sum_{j'=j}^{i} w_{\pi_{j'}} - cost(j,i), & \text{if } j \leq i; \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

$F[0][0] = 0$ is the boundary condition and we need to compute $F[m][n]$. By tracing back the optimal substructures, we are able to identify m vehicle routes and the profit p_k for each vehicle k. Note that $p_k \geq F[m][n], \forall\, k \in V$ and the MRP is maximized. We then apply the model M_2 in Section IV-B1 to reassign vehicle routes to transport companies.

4) Recombination: The recombination process produces a new set P' of solutions by mating two individual solutions in P. It iteratively selects two solutions that has not been selected from P and mates them with a probability of p_m, which is a user-defined parameter. First, each of the selected solutions is converted to a giant tour. Subsequently, two giant tours are recombined to generate two new giant tours. Since the giant tour is a permutation of distinct elements, any order-based crossover operator can be adopted for recombination. In our implementation, we use the Partially Mapped Crossover (PMX) operator [37]. Next, two splitting methods are respectively applied to each of the new giant tours, generating a total of four solutions. Finally, the solution that is not dominated by the other three is added in the population set P'.

Algorithm 3 The Large Neighborhood Search Procedure

1: $non\_impr \leftarrow 0$;
2: **repeat**
3:     $s' \leftarrow$ **destroy_repair**(s);
4:     **if** $s' \prec s$ **then**
5:         $s \leftarrow s'$;
6:         $non\_impr \leftarrow 0$;
7:     **else if** $s \preceq s'$ **then**
8:         $non\_impr \leftarrow non\_impr + 1$;
9:     **else**
10:         // s and s' are not comparable, nothing to do here;
11:     **end if**
12: **until** non_impr reaches a predefined value λ.

5) Large Neighborhood Search: The large neighborhood search (LNS) heuristic was proposed by Shaw [38]. The neighborhood is defined by a *destroy* and *repair* method. A destroy method destructs part of the current solution to a partial solution while a repair method rebuilds the destroyed solution to a complete one. Therefore, the neighborhood of a solution is the set of solutions that can be reached by a destroy operation and a repair operation.

Pseudocode for the LNS heuristic is shown in Algorithm 3. *non_impr* is a parameter recording the number of consecutive iterations that cannot improve on the current best solution. The destroy-and-repair operator is applied to generate a new solution s' (see line 3, Algorithm 3). Different from the local search procedure that a solution is accepted if the k-th objective gets improved, the LNS procedure only accepts s' when $s' \prec s$ (see lines 4 – 5, Algorithm 3). The LNS heuristic can generate a large number of solutions that are not comparable with s but are very likely to be dominated by other solutions in $P \cup P'$. If s' is dominated by or equal to s, *non_impr* is increased by 1 (see line 8, Algorithm 3). If s and s' cannot dominate each other, the algorithm does nothing (see line 10, Algorithm 3). The LNS heuristic terminates when *non_impr* reaches a predefined value λ.

The destroy method removes a percentage of customers from s. Although several adaptive LNS heuristics were proposed [39], we employ a simple destroy method that randomly removes each customer with a probability of p_r. Then, the removed customers are inserted in the partial solution using a repair method, which works as follows. First, the removed customers are sorted by descending demands. Next, these customers are sequentially inserted into the partial solution. Since there are two objectives, we introduce two insertion criteria, namely *Cost-oriented insertion* and *Fairness-oriented insertion*. Note that the insertion that respects the vehicle

TABLE I
PARAMETER SETTINGS

Algorithm	Parameter	Value
MOLS & MOLS+	Population size ($\|P\|$)	200
	The maximum number of generations (N_{gen})	10000
	The maximum number of iterations (*Iter*)	3000
	The coefficient for tournament size (τ)	0.5
MOLS+	Recombination probability (p_m)	1.0
	The maximum number of non-improving iterations (λ)	10
	Removal probability (p_r)	0.1

capacity constraint is allowed. The destroy-and-repair process randomly chooses these two insertion criteria whose details are as follows:

1) *Cost-oriented insertion*. Let $\Delta g_1(k, j)$ be the change of the total traveling cost resulted by inserting the customer at position j in route k. Perform the insertion associated with (k, j) that is determined by:

$$(k, j) := \arg \min_{k \in V, j \in r_k} \Delta g_1(k, j) \tag{29}$$

2) *Fairness-oriented insertion*. Let $g_2(t)$ be the unit profit of transport company t and $\Delta g_2(t, k, j)$ be the change of unit profit of transport company t if the customer is inserted at position j in route k. Therefore, $g_2(t) + \Delta g_2(t, k, j)$ is the resultant unit profit of transport company t. Expression (30) finds a (t, k, j)-tuple that makes the minimum unit profit maximal. The customer is then inserted at position j in route k.

$$(t, k, j) := \arg \max_{t,k,j} \left\{ \min_{t \in T, k \in V(t), j \in r_k} g_2(t) + \Delta g_2(t, k, j) \right\}. \tag{30}$$

V. COMPUTATIONAL RESULTS

Our algorithms were coded in GNU C++ with the "-O2" option. All experiments were conducted on a Linux server equipped with an Intel Xeon E5-1603 CPU clocked at 2.80 gigahertz and 8 gigabytes RAM. The parameters in MOLS and MOLS+ were carefully calibrated based on some preliminary experiments. The final parameter settings are presented in Table I. For each instance, we executed each algorithm 10 independent runs with different random seeds. All computation times reported here are in CPU seconds on this server. All instances and detailed results are available upon request.

The first experiment was to compare MOLS+ with an existing algorithm. As the VRPOPB is a new problem, there are not existing benchmark instances. We find in the literature that the VRP with routing balancing (VRPRB) introduced by Jozefowiez *et al.* [13] is very similar to the VRPOPB. The VRPRB has two objectives, namely, minimizing the total traveling distance and minimizing the difference between the longest and the shortest route lengths (note that this objective may lead to distorted solutions). Jozefowiez *et al.* [13] proposed a meta-heuristic method based on a parallel evolutionary algorithm and used seven instances obtained from Christofides and Eilon [40] to evaluate their method. For each

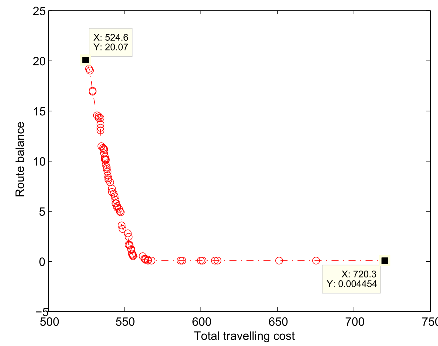


Fig. 3. The approximate Pareto-optimal set containing 91 solutions produced by MOLS+ for instance E51-05e.

of these instances, they reported the best-known value for the total traveling distance, the best found value for the total traveling distance as well as the associated route balance, and the best found route balance as well as the associated total length. To compare with this method, we modified our MOLS+ to solve the VRPRB by simply changing the second objective of the VRPOPB into minimizing the difference between the longest and the shortest route lengths.

The results of the first experiment are shown in Table II. The name of each instance has the form $Ei - jk$, where E indicates that the distance metric is Euclidean, i is the number of customers, j is the number of vehicles and k is an identifier. As can be seen, the numbers of customers and vehicles range from 51 to 200 and from 5 to 17, respectively. The column "Best-known" gives the optimal total traveling distance found by existing VRP approaches. Each of these two approaches can generate a set of non-dominated solutions, but only two extreme solutions with respect to different primary objectives are presented in this table. Under the block "Solely minimizing the total traveling distance", we mark in bold the best found value that is equal to the best-known value. After comparing columns 3 and 6, we can see that MOLS+ produced the total traveling distance shorter or equal to that found by Jozefowiez *et al.* [13]'s method for every instances. Columns 5 and 8 give the running times of two algorithms. Columns 9 and 11 show that MOLS+ generated much more balanced solutions than Jozefowiez *et al.* [13]'s method. The numbers under "Best found" columns indicate the route difference, i.e., the difference between the longest and shortest route. Moreover, under the block "Solely minimizing the difference", MOLS+ obtained solutions which dominate those produced by Jozefowiez *et al.* [13]'s method for instances E101-08e and E101-10c completely and significantly. We can judge from this table that MOLS+ is able to produce high quality solutions for the VRPRB.

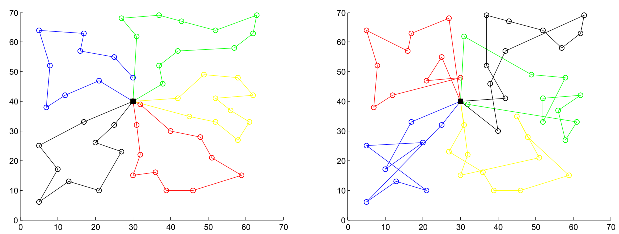
We also draw in Figure 3 the approximate Pareto-optimal set containing 91 solutions produced by MOLS+ for instance E51-05e. However, after taking a close look at those balanced solutions (e.g., see Figure 4(b)), many routes take detours, and thus the solutions are actually distorted.

The second experiment was conducted on 57 restricted VRPOPB instances in which each transport company only owns one vehicle. We generated these restricted VRPOPB instances based on 7 VRPRB instances from

TABLE II

PERFORMANCE COMPARISON BETWEEN JOZEFOWIEZ *et al.* [13]'S METHOD AND MOLS+ ON VRPRB INSTANCES

Instance	Solely minimizing the total traveling distance							Solely minimizing the difference			
	Best known	Jozefowiez et al. [13]'s method			MOLS+			Jozefowiez et al. [13]'s method		MOLS+	
		Best found	Associated balance	Time	Best found	Associated balance	Time	Best found	Associated length	Best found	Associated length
E51-05e	524.61	**524.61**	20.07	613.2	**524.61**	20.07	905.7	0.24	618.22	0.004	720.28
E76-10e	835.26	835.32	78.10	1522.8	835.32	78.10	1186.1	0.59	1203.98	0.047	1654.80
E101-08e	826.14	827.39	67.55	2113.2	**826.14**	97.89	1643.9	0.29	1871.06	**0.009**	**1358.40**
E101-10c	819.56	**819.56**	93.43	2125.8	**819.56**	93.43	1738.6	1.15	1429.9	**0.018**	**1272.88**
E121-07c	1042.11	**1042.11**	146.67	2418.0	**1042.11**	146.67	1775.1	0.1	2388.3	0.004	3037.22
E151-12c	1028.42	1047.35	74.78	3936.0	1040.74	79.63	2551.2	0.8	1484.48	0.046	1913.89
E200-17c	1291.45	1352.46	76.60	4932.0	1331.06	90.70	3566.3	1.38	1902.64	0.109	2161.93



(a) (b)

Fig. 4. (a) The solution for instance E51-05e with the minimum traveling distance. (b) the solution for instance E51-05e with the smallest difference between the longest and the shortest route lengths.

Jozefowiez *et al.* [13] and 50 VRP instances from Augerat *et al.* [41] as follows. The reward w_i is set to 0 for each $i \in N$. Each transport company $t \in T$ has only one vehicle. The traveling cost is set according to the distance matrix. Other inputs (e.g., the number of vehicles, customer demands) are the same as those of the original instances. Therefore, the second objective becomes minimizing the maximum route length among m routes. In order to evaluate the performance of MOLS and MOLS+, the following two indicators are employed.

- *Inverted generational distance.* The inverted generational distance (IGD) indicator is used to measure how "far" an approximate Pareto front is away from the true Pareto front [42]. Given a set A of non-dominated solutions and a reference set P^*, $IGD(A, P^*)$ is defined as:

$$IGD(A, P^*) = \frac{1}{|P^*|} \sum_{y \in P^*} min_{x \in A}\ d(x, y) \tag{31}$$

In this formula, $d(x, y)$ corresponds to the distance between the solutions x and y, which is calculated by:

$$\begin{aligned} d(x, y) &= \sqrt{\sum_{i=1}^{2} \left(\frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}} - \frac{f_i(y) - f_i^{min}}{f_i^{max} - f_i^{min}} \right)^2} \\ &= \sqrt{\sum_{i=1}^{2} \left(\frac{f_i(x) - f_i(y)}{f_i^{max} - f_i^{min}} \right)^2}, \end{aligned} \tag{32}$$

where f_i^{max} and f_i^{min} are the maximum and minimum values of the i-th objective among the solutions in P^*. Since the true Pareto front is always difficult to be obtained, in our experiment we set the reference set P^* to the Pareto front of all solutions obtained by MOLS and MOLS+. A small $IGD(A, P^*)$ indicates that front A is a good approximation for the reference set P^*.

- *Hypervolume.* The hypervolume (HV) indicator was first proposed by Zitzler and Thiele [43]. We calculate the value of the hypervolume indicator between the approximate Pareto front and a reference point. For the minimization problem, the reference point z is composed of the maximum values of all objectives. For a two-dimensional objective space, each non-dominated solution covers a rectangle, which is defined by the lower-left coordinate $(f_1(s), f_2(s))$ and the upper-right reference point $z = (z_1, z_2)$. The HV value is given by the area of the union of all rectangles covered by the Pareto-optimal solutions. The HV value can reflect both the convergence and diversity of the non-dominated solutions. The larger the HV value is, the closer the corresponding non-dominated solutions are to the Pareto front.

MOLS+ produced significantly better results than MOLS according to the second experiment, where the impact of each component of MOLS+ was also studied. The first component of MOLS+ is the giant-tour representation with dynamic programming splitting, and the second one is the large neighborhood search procedure. If we disable the first component of MOLS+, the resultant approach is referred to as MOLS+LNS. Similarly, MOLS+GT corresponds to the approach without the second component. Table III shows the IGD values, the HV values and the running times achieved by the four algorithms for 7 restricted VRPOPB instances, where the best IGD and HV values are marked in bold. For the IGD value, MOLS+LNS, MOLS+GT, MOLS+ improved on MOLS by 68.1%, 79.9% and 95.0%, respectively. For the HV value, the corresponding improvements are 40.2%, 41.4% and 46.5%, respectively. On average the running time of MOLS+ is around 3 times as much as that of MOLS.

Table IV presents the best objective value found by MOLS and MOLS+ for each of the 57 restricted VRPOPB instances. The total traveling costs that are equal to their corresponding best-known values are marked in bold. Note that the rewards of all customers are zero, so the profit of each instance must be a negative value. For the first objective, MOLS+ reached all best-known results except for 4 instances, namely, E151-12c, E200-17c, A-n63-k9 and A-n64-k9. On average, MOLS (resp. MOLS+) produced solutions with the total traveling cost increasing from 1025.2 (resp. 994.7) to 1143.9 (resp. 1095.9) and with the smallest profit increasing from −207.2 (resp. −204.6) to −180.5 (resp. −178.8). As can be seen, MOLS+ produced more satisfactory results for these instances. The running time of MOLS+ is about twice longer than that of MOLS.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
THE PERFORMANCE COMPARISON OF FOUR ALGORITHMS ON 7 RESTRICTED VRPOPB INSTANCES

Instance	MOLS			MOLS + LNS			MOLS + GT			MOLS+		
	IGD	HV	Time	IGD	HV	Time	IGD	HV	Time	IGD	HV	Time
E051-05e	0.4040	0.4909	535.4	0.0683	0.8810	818.2	0.0217	0.9310	627.0	**0.0034**	**0.9620**	918.5
E076-10e	0.4623	0.3793	587.2	0.1733	0.8770	1017.5	0.1636	0.7850	722.4	**0.0202**	**0.9575**	1172.9
E101-08e	0.3180	0.4938	632.4	0.1099	0.7859	1384.9	0.0561	0.8253	843.3	**0.0206**	**0.9031**	1626.2
E101-10c	0.1573	0.7473	652.3	0.0833	0.8989	1422.6	0.0178	0.9513	898.0	**0.0103**	**0.9586**	1676.2
E121-07c	0.3924	0.3185	574.8	0.1309	0.5980	1478.3	0.0974	0.6585	840.7	**0.0354**	**0.7749**	1744.5
E151-12c	0.3242	0.5389	683.2	0.1148	0.8547	1959.3	0.0711	0.8563	1130.5	**0.0174**	**0.9456**	2482.5
E200-17c	0.4250	0.4951	706.2	0.1119	0.8919	2516.9	0.0712	0.9075	1502.7	**0.0171**	**0.9669**	3337.0
Average	0.3547	0.4948	624.5	0.1132	0.8268	1513.9	0.0713	0.8450	937.8	**0.0178**	**0.9241**	1851.1

TABLE IV
PERFORMANCE COMPARISON BETWEEN MOLS AND MOLS+ ON 57 RESTRICTED VRPOPB INSTANCES

Instance	Solely minimizing the total traveling cost					Solely maximizing the minimum profit				Running time	
	Best-known	MOLS		MOLS+		MOLS		MOLS+		MOLS	MOLS+
		Best found	Associated minimum profit	Best found	Associated minimum profit	Best found	Associated traveling cost	Best found	Associated traveling cost		
E51-05e	524.61	531.02	-115.52	**524.61**	-118.52	-111.37	537.89	-111.37	537.89	535.5	918.5
E76-10e	835.26	863.34	-106.59	**835.26**	-120.16	-97.00	930.17	-93.03	888.10	587.2	1172.9
E101-08e	826.14	843.13	-131.56	**826.14**	-138.79	-115.53	900.70	-110.42	859.46	632.4	1626.2
E101-10c	819.56	853.47	-139.88	**819.56**	-137.02	-120.81	1070.86	-120.72	934.43	652.3	1676.2
E121-07c	1042.11	1197.31	-224.45	**1042.12**	-213.63	-204.90	1344.69	-199.94	1299.10	574.8	1744.5
E151-12c	1028.42	1084.60	-120.35	1031.88	-120.62	-103.52	1185.23	-100.75	1099.85	683.2	2481.5
E200-17c	1291.45	1385.93	-114.07	1308.53	-115.09	-99.98	1633.83	-99.86	1411.11	706.2	3336.0
A-n32-k5	784	**784**	-267	**784**	-267	-209	948	-209	926	280.4	411.6
A-n33-k5	661	**661**	-185	**661**	-185	-157	736	-157	728	277.1	407.1
A-n33-k6	742	743	-173	**742**	-172	-150	860	-148	861	272.6	407.6
A-n34-k5	778	**778**	-188	**778**	-188	-173	807	-173	806	269.6	403.7
A-n36-k5	799	814	-255	**799**	-226	-213	936	-213	881	278.5	414.9
A-n37-k5	669	**669**	-211	**669**	-211	-152	740	-152	739	285.4	430.1
A-n37-k6	949	961	-278	**949**	-250	-210	1068	-209	1077	268.3	416.3
A-n38-k5	730	731	-210	**730**	-184	-164	785	-164	785	261.2	407.3
A-n39-k5	822	828	-212	**822**	-212	-198	909	-196	937	266.8	416.7
A-n39-k6	831	833	-209	**831**	-221	-181	991	-179	1012	283.7	435.1
A-n44-k7	937	955	-254	**937**	-248	-198	1190	-198	1081	265.5	438.9
A-n45-k6	944	1008	-204	**944**	-204	-203	1056	-193	1010	259.4	420.0
A-n45-k7	1146	1180	-224	**1146**	-221	-205	1322	-202	1303	288.4	457.5
A-n46-k7	914	923	-195	**914**	-198	-181	1035	-180	1045	290.8	464.7
A-n48-k7	1073	1085	-206	**1073**	-206	-200	1272	-196	1285	289.0	471.9
A-n53-k7	1010	1041	-208	**1010**	-225	-197	1225	-196	1121	280.2	488.8
A-n54-k7	1167	1186	-220	**1167**	-228	-202	1251	-201	1239	276.4	485.3
A-n55-k9	1073	1074	-176	**1073**	-176	-161	1215	-161	1175	295.1	512.4
A-n60-k9	1354	1387	-230	**1354**	-230	-223	1669	-223	1478	297.2	544.4
A-n61-k9	1034	1142	-183	**1034**	-173	-158	1299	-147	1124	290.3	500.4
A-n62-k8	1288	1318	-241	**1288**	-252	-212	1557	-210	1460	292.7	547.1
A-n63-k10	1314	1339	-192	**1314**	-204	-187	1685	-187	1416	295.0	557.8
A-n63-k9	1616	1664	-266	1618	-267	-245	1834	-244	1725	285.3	529.7
A-n64-k9	1401	1428	-257	1409	-258	-219	1695	-217	1546	294.3	554.3
A-n65-k9	1174	1230	-206	**1174**	-181	-172	1308	-164	1213	288.1	534.7
A-n69-k9	1159	1191	-205	**1159**	-171	-148	1283	-146	1224	296.4	584.8
A-n80-k10	1763	1858	-268	**1763**	-288	-252	2088	-252	1889	298.6	648.2
B-n31-k5	672	**672**	-189	**672**	-189	-189	672	-189	672	282.9	391.8
B-n34-k5	788	**788**	-212	**788**	-212	-198	868	-194	825	270.1	403.8
B-n35-k5	955	**955**	-247	**955**	-247	-233	1049	-233	1049	279.9	412.3
B-n38-k6	805	**805**	-217	**805**	-211	-174	916	-174	912	286.6	433.1
B-n39-k5	549	**549**	-196	**549**	-196	-190	649	-188	805	278.3	432.7
B-n41-k6	829	836	-187	**829**	-187	-169	868	-169	866	268.2	424.4
B-n43-k6	742	**742**	-190	**742**	-182	-155	831	-155	806	287.1	452.2
B-n44-k7	909	910	-212	**909**	-212	-167	986	-167	985	278.9	446.7
B-n45-k5	751	769	-223	**751**	-215	-187	849	-184	860	263.9	434.1
B-n45-k6	678	713	-156	**678**	-156	-143	781	-143	711	263.5	416.4
B-n50-k7	741	**741**	-151	**741**	-151	-141	827	-141	814	292.6	492.9
B-n50-k8	1312	1325	-243	**1312**	-243	-226	1466	-226	1436	283.8	485.4
B-n51-k7	1032	1042	-221	**1032**	-215	-193	1178	-190	1134	266.9	454.0
B-n52-k7	747	752	-222	**747**	-214	-151	792	-151	784	295.6	505.6
B-n56-k7	707	708	-202	**707**	-202	-182	763	-182	743	294.1	515.5
B-n57-k7	1153	1482	-288	**1153**	-202	-197	1236	-199	1193	286.9	462.0
B-n57-k9	1598	1608	-251	**1598**	-249	-223	1788	-222	1659	302.0	532.0
B-n63-k10	1496	1560	-215	**1496**	-238	-212	1765	-210	1668	295.0	555.0
B-n64-k9	861	932	-188	**861**	-204	-168	1025	-168	883	282.1	527.7
B-n66-k9	1316	1330	-275	**1316**	-267	-204	1541	-200	1525	288.7	558.1
B-n67-k10	1032	1071	-213	**1032**	-237	-194	1123	-184	1151	300.7	584.2
B-n68-k9	1272	1294	-215	**1272**	-207	-188	1367	-187	1425	297.1	583.0
B-n78-k10	1221	1283	-221	**1221**	-218	-182	1497	-181	1446	295.8	634.9
Avg.	994.1	1025.2	-207.2	994.7	-204.6	-180.5	1143.9	-178.8	1095.9	325.2	648.9

We also plot in Figure 5 the non-dominated solutions found by our MOLS+ for instance E51-05e. There are 5 non-dominated solutions in total. The best solutions corresponding to optimizing the first and the second objectives are given by the extreme points (524.6, −118.5) and (537.9, −111.4), respectively. The detailed solutions related to these two extreme points are pictorially shown in Figure 6. It is easy to find that the routes in Figure 6(a) are the same as those in Figure 4(b). Figure 6(b) shows a solution with balanced profits in which each route contains no detour. The route drawn in black has the smallest profit (i.e., −111.4). The result is consistent with Theorem 1 that no distorted solution is contained in the Pareto front.

The third experiment is similar to the second one. We modified 57 restricted VRPOPB instances to generate a set of VRPOPB instances as follows. The reward w_i is set to $2 \cdot max_{j \in N}\{c_{ij}\}$ for $i \in N$, which ensures that the profit gained by a vehicle must be positive. If m vehicles are available, the number of transport companies $|T|$ is set to $\lceil \sqrt{m} \rceil$. Each of the first ($m \bmod |T|$) transport companies has $\lceil m/|T| \rceil$ vehicles and each of the remaining companies has $\lfloor m/|T| \rfloor$ vehicles. For example, if $m = 5$, then $|T| = 3$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

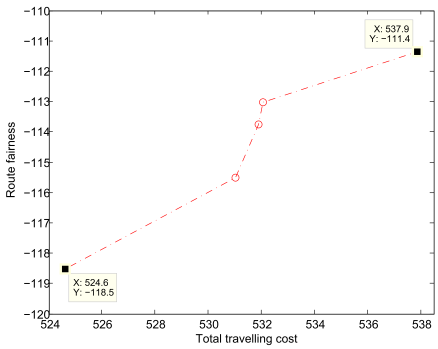
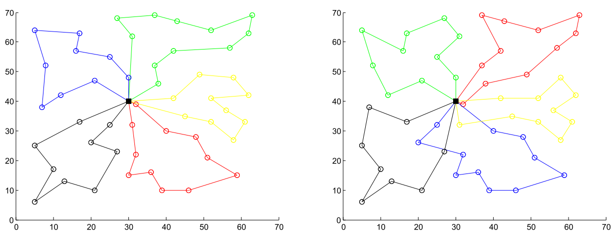


Fig. 5. The five non-dominated solutions found by MOLS+ for instance E51-05e.



(a) (b)

Fig. 6. (a) The solution for instance E51-05e with the minimum traveling cost. (b) the solution for instance E51-05e with the balanced profits.

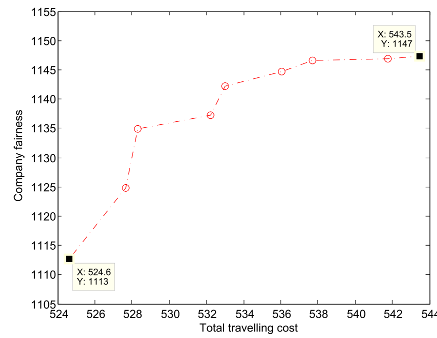
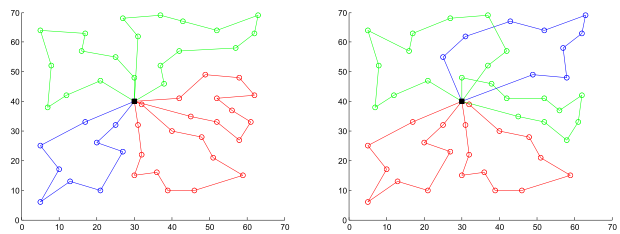


Fig. 7. The nine non-dominated solutions found by MOLS+ for instance E51-05e.



(a) (b)

Fig. 8. (a) The solution for instance E51-05e with the minimum traveling cost. (b) the solution for instance E51-05e with the balanced profits.

TABLE V
THE RESULTS OF MOLS AND MOLS+ ON 7 VRPOPB INSTANCES

Instance	MOLS				MOLS+			
	NS	IGD	HV	Time	NS	IGD	HV	Time
E051-05e	4.6	0.4017	0.5636	543.7	8.4	0.0061	0.9905	925.7
E076-10e	4.2	0.4454	0.5164	604.5	10.5	0.0351	0.9756	1215.8
E101-08e	5.4	0.4790	0.5346	655.6	8.3	0.0129	0.9863	1715.1
E101-10c	4.0	0.3233	0.6703	683.6	8.1	0.0200	0.9911	1794.5
E121-07c	4.4	0.5347	0.4379	609.2	10.7	0.1668	0.8116	1779.1
E151-12c	5.8	0.3334	0.6722	721.5	16.7	0.0474	0.9879	2603.4
E200-17c	7.4	0.3123	0.6432	742.6	18.8	0.0417	0.9820	3548.2
Average	5.1	0.4043	0.5769	651.5	11.6	0.0471	0.9607	1940.3

and three transport companies have 2, 2 and 1 vehicles, respectively.

Table V reports the number of non-dominated solutions (NS), IGD, HV and running time obtained by MOLS and MOLS+ for each of the 7 VRPOPB instances, where the number in each cell is the average value over 10 runs. From this table, we can see that the number of non-dominated solutions found by MOLS+ is around twice as many as that obtained by MOLS. The IGD and HV values both obviously demonstrate the superiority of MOLS+.

Table VI gives two extreme solutions achieved by MOLS and MOLS+ for each of 57 instance. MOLS+ reached 51 out of 57 best-known solutions when solely considering the first objective. From the results produced by MOLS+, we can see that on average the total traveling cost increases from 995.4 to 1010.8 and the smallest unit profit increases from 1166.7 to 1221.4. This means that an increase of around 1.5% in traveling cost leads to a 4.5% profit increase for the company with least gain. It may appear that such trade-off is not significant due to the nature of the *max_min* fairness function. We then included the conventional difference function (i.e., maximum unit profit minus minimum unit profit) and presented the corresponding results in columns 5 and 8. Under this fairness function, increasing 1.5% traveling cost results in decreasing 97.7% (=(124.4-2.9)/124.4 *100%) profit difference. This function reveals the importance of objective trade-off between optimizing the traveling cost and profit balancing.

Figure 7 shows 9 non-dominated solutions found by MOLS+ for instance E51-05e. The lower-left point (524.6,1113) and the upper-right point (543.5,1147) correspond to the solutions in Figures 8(a) and 8(b), respectively. This instance includes 5 vehicles owned by 3 transport companies. The routes in the same color belong to the same transport company.

VI. CASE STUDY

We study a real case from a food importer in Hong Kong distributing assorted food products (e.g., rice and flour) to stores. We have collected a large number of historical orders placed in a month. For each day in this month, the vehicle schedule has already been made and implemented to fulfill the corresponding orders. We arbitrarily chose the orders of one day for this case study. After a basic data cleaning phase, we have a total of 472 orders, each of which can be simply characterized by its address and demand. The average demand of these orders is 6.8 *basic transportation units* (BTUs). The locations of these orders are shown in Figure 9, where the warehouse of the food importer is located in *Yuen Long* Industrial Estate (see the marker in green at the upper-left corner of the map). The food importer has employed

TABLE VI

PERFORMANCE COMPARISON BETWEEN MOLS AND MOLS+ ON 57 VRPOPB INSTANCES

Instance	Solely minimizing the total traveling cost							Solely maximizing the minimum unit profit						Running time	
	Best known	MOLS			MOLS+			MOLS			MOLS+			MOLS	MOLS+
		Best found	Associated minimun unit profit	Associated profit difference	Best found	Associated minimum unit profit	Associated profit difference	Best found	Associated total traveling cost	Associated profit difference	Best found	Associated total traveling cost	Associated profit difference		
E051-05e	524.61	538.99	1112.65	62.32	**524.61**	1112.65	67.27	1140.40	575.24	7.59	1147.28	543.47	2.92	543.7	925.7
E076-10e	835.26	872.11	833.99	149.29	**835.26**	859.06	61.52	874.51	942.84	3.35	884.96	841.21	3.57	604.5	1215.8
E101-08e	826.14	851.35	1488.40	250.69	827.39	1614.40	35.17	1626.10	889.35	0.28	1632.93	832.71	0.99	655.6	1715.1
E101-10c	819.56	859.68	1330.15	179.94	**819.56**	1416.54	13.57	1403.43	985.69	1.24	1418.05	828.82	4.05	683.6	1794.5
E121-07c	1042.11	1250.48	2950.41	400.13	1071.96	3092.82	32.85	3073.21	1329.27	0.96	3108.69	1077.76	2.22	609.2	1779.1
E151-12c	1028.42	1076.02	1490.63	266.64	1040.25	1403.01	419.68	1624.52	1156.17	0.56	1633.27	1050.03	0.75	721.5	2602.5
E200-17c	1291.45	1390.09	1315.81	456.16	1311.66	1322.37	376.59	1505.74	1596.51	0.17	1521.43	1324.28	1.17	742.6	3547.3
A-n32-k5	784	**784**	974.5	444.5	**784**	1041.0	199.0	1070.0	856	10.5	1080.5	814	11.5	552.7	788.8
A-n33-k5	661	**661**	815.0	341.0	**661**	948.0	141.5	1039.0	727	5.0	1044.0	701	6.5	541.0	786.4
A-n33-k6	742	**742**	763.0	141.5	**742**	810.5	24.0	822.0	754	4.5	822.0	753	4.5	552.1	797.2
A-n34-k5	778	**778**	1064.0	59.0	**778**	1064.0	59.0	1084.0	785	13.5	1087.0	796	2.0	530.9	797.5
A-n36-k5	799	812	742.0	647.0	**799**	1124.0	144.0	1177.0	853	2.0	1180.0	817	12.5	562.1	810.3
A-n37-k5	669	**669**	802.0	723.5	**669**	1074.5	271.5	1158.5	744	6.0	1168.0	698	9.0	554.1	847.2
A-n37-k6	949	967	782.0	371.5	**949**	866.0	248.0	950.5	1012	5.0	959.0	969	2.0	535.5	812.0
A-n38-k5	730	736	1077.0	435.0	**730**	1103.0	208.5	1208.0	773	9.5	1215.0	751	8.5	513.8	802.5
A-n39-k5	822	826	1222.0	87.0	**822**	1243.5	46.0	1260.0	830	9.0	1261.5	833	6.5	526.2	820.4
A-n39-k6	831	833	868.5	409.0	**831**	960.5	351.5	1110.5	936	5.0	1123.0	865	5.0	553.4	860.3
A-n44-k7	937	943	675.0	835.0	**937**	1008.0	101.0	1030.7	1006	6.3	1039.5	947	8.0	536.1	927.8
A-n45-k6	944	994	1253.0	252.5	**944**	1335.5	108.5	1361.5	1016	10.5	1372.5	970	4.0	506.9	835.5
A-n45-k7	1146	1161	872.0	143.5	**1146**	944.5	14.5	946.0	1180	1.3	950.0	1156	0.5	577.2	921.6
A-n46-k7	914	917	634.0	970.0	**914**	1052.5	43.0	1057.0	968	1.0	1062.5	931	0.5	589.2	922.6
A-n48-k7	1073	1090	1100.0	79.0	**1073**	1106.0	45.5	1122.7	1132	9.3	1129.0	1105	1.5	583.0	941.8
A-n53-k7	1010	1055	801.5	809.5	**1010**	876.0	745.0	1255.0	1092	1.0	1264.0	1020	4.5	560.3	980.5
A-n54-k7	1167	1186	1187.0	176.3	**1167**	1276.0	32.0	1284.0	1194	10.0	1289.7	1179	2.3	555.4	978.2
A-n55-k9	1073	1101	875.3	281.7	**1073**	1015.3	10.7	1011.3	1129	4.3	1017.7	1085	2.0	610.2	1056.7
A-n60-k9	1354	1380	1086.0	110.3	**1354**	1114.3	37.0	1127.0	1418	1.7	1134.0	1361	0.7	626.1	1149.1
A-n61-k9	1034	1125	720.0	530.0	**1034**	975.3	121.7	999.7	1241	4.7	1023.0	1050	0.3	585.4	1045.3
A-n62-k8	1288	1314	1061.3	568.0	**1288**	1344.3	64.2	1360.3	1376	10.7	1371.7	1304	1.3	608.7	1138.9
A-n63-k10	1314	1344	898.7	380.8	**1314**	974.0	142.7	1034.0	1477	4.3	1051.5	1325	0.8	612.9	1151.4
A-n63-k9	1616	1664	952.0	495.3	**1616**	1177.7	7.3	1163.0	1772	1.7	1179.0	1633	0.7	595.0	1106.8
A-n64-k9	1401	1466	718.3	828.0	1403	1053.3	127.7	1085.3	1539	2.0	1098.7	1417	4.0	617.5	1169.4
A-n65-k9	1174	1197	1101.3	222.3	**1174**	1211.3	26.0	1215.0	1247	0.7	1221.0	1179	5.3	585.6	1111.9
A-n69-k9	1159	1209	1266.3	135.7	**1159**	1206.3	211.0	1331.3	1238	1.3	1339.3	1169	0.3	626.3	1227.8
A-n80-k10	1763	1845	1096.5	929.5	**1763**	1401.0	74.5	1403.0	2022	11.5	1428.5	1788	1.5	616.7	1352.8
B-n31-k5	672	675	872.5	1.0	**672**	855.5	42.0	872.5	675	1.0	872.5	675	1.0	551.9	782.3
B-n34-k5	788	**788**	758.0	619.0	**788**	953.0	138.0	996.5	810	17.5	1003.0	812	8.0	537.8	799.6
B-n35-k5	955	**955**	923.0	650.5	**955**	1162.0	94.0	1188.0	978	15.0	1195.5	975	0.5	549.9	806.2
B-n38-k6	805	806	786.0	398.0	**805**	947.0	57.5	978.0	823	3.0	980.0	820	1.0	566.9	850.9
B-n39-k5	549	550	1080.5	804.5	**549**	1176.0	452.0	1436.5	667	2.0	1450.5	588	5.5	554.4	859.7
B-n41-k6	829	831	1185.5	86.0	**829**	1164.0	107.5	1205.0	869	9.0	1211.0	856	0.0	530.3	841.6
B-n43-k6	742	745	844.5	162.5	**742**	875.5	125.0	926.0	757	3.5	927.5	749	3.5	564.2	888.3
B-n44-k7	909	927	818.7	85.8	**909**	830.0	60.5	845.0	975	2.0	852.5	920	4.0	563.3	889.2
B-n45-k5	751	753	1411.5	542.5	**751**	1565.0	50.0	1572.5	791	4.0	1580.0	760	1.5	514.7	844.9
B-n45-k6	678	736	884.5	394.0	**678**	1049.0	93.0	1075.0	788	0.0	1088.0	706	1.5	510.1	831.7
B-n50-k7	741	**741**	978.3	615.2	**741**	1113.0	137.0	1153.3	762	0.2	1155.0	751	0.0	599.6	994.3
B-n50-k8	1312	1327	742.3	289.0	**1312**	876.3	170.7	919.5	1346	0.5	923.0	1316	0.7	605.0	993.6
B-n51-k7	1032	1050	1442.0	335.0	**1032**	1548.7	25.8	1543.5	1116	1.8	1553.3	1052	0.2	531.1	906.7
B-n52-k7	747	753	1172.5	22.0	**747**	1168.5	60.0	1184.7	770	2.8	1186.7	750	5.8	606.5	1014.2
B-n56-k7	707	715	1382.3	199.2	**707**	1442.5	56.5	1454.7	766	3.3	1461.3	722	1.2	602.3	1037.8
B-n57-k7	1153	1383	1468.7	348.3	**1153**	1639.5	73.2	1645.0	1410	0.5	1678.0	1177	0.7	539.0	937.9
B-n57-k9	1598	1636	916.7	90.0	**1598**	894.0	204.3	965.3	1641	0.3	969.0	1605	0.7	641.4	1113.4
B-n63-k10	1496	1542	1013.0	319.0	1503	1103.3	143.2	1126.7	1671	0.8	1141.0	1527	2.0	617.4	1137.0
B-n64-k9	861	939	1100.0	185.0	**861**	1198.0	3.0	1188.7	951	2.0	1198.3	863	2.7	585.2	1099.2
B-n66-k9	1316	1335	944.0	479.3	**1316**	1185.3	53.0	1201.3	1360	1.3	1205.7	1324	0.3	609.6	1159.2
B-n67-k10	1032	1079	781.5	717.2	**1032**	1222.0	78.5	1231.0	1161	4.5	1242.7	1051	1.3	638.8	1197.3
B-n68-k9	1272	1298	1178.0	33.3	**1272**	1142.7	175.3	1195.7	1321	1.7	1201.0	1274	1.3	628.6	1212.8
B-n78-k10	1221	1250	1178.7	211.8	**1221**	1264.0	79.0	1274.5	1350	2.8	1287.0	1226	2.0	623.0	1306.6
Average	994.1	1025.5	1049.0	364.2	995.4	1166.7	124.4	1212.2	1079.8	4.3	1221.4	1010.8	2.9	583.3	1091.7

TABLE VII

INFORMATION OF THE REAL CASE

Number of the orders	472
Number of the transport companies	9
Number of the vehicles	20 (=3+2+2+1+3+3+2+3+1)
Average demand	6.8
Vehicle capacity	≤ 250

20 vehicles each with a capacity of less than 250 BTUs from 9 transport companies to deliver food products. The scale of a transport company is set to the number of vehicle it maintains. We summarize the above information in Table VII.

Currently, a seasoned scheduler of the food importer is in charge of assigning orders to vehicles and planning the vehicle routes. The manual planning basically adopts the *clustering* strategy. Hong Kong has three main regions, namely, New Territories (in the north), Kowloon (in the middle) and Hong Kong Island (in the south). A vehicle is dispatched to carry out the orders whose locations are adjacent to each other in a particular region. Figure 10 illustrates the vehicle assignment made by the scheduler for the orders in New Territories and Hong Kong Island. The number inside the marker indicates the vehicle to which the corresponding order is assigned.

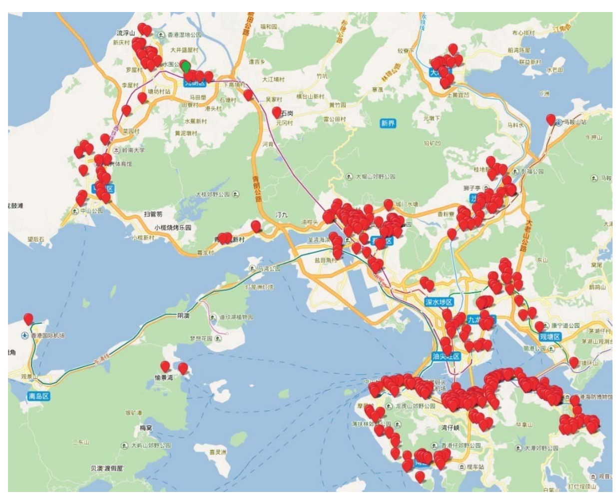


Fig. 9. Distribution of orders across Hong Kong.

The *clustering* strategy is easy to be implemented, but it leads to large traveling cost and profit unbalancing. Some clusters may contain only a few number of orders, resulting in vehicle routes with light loads and low rewards. We applied our MOLS+ to this real case and obtained a set

TABLE VIII

THE COMPARISON BETWEEN THE MOST BALANCED SOLUTION GENERATED BY MOLS+ AND THE MANUALLY GENERATED SOLUTION

Company	Solution from the scheduler				Balanced solution found by MOLS+			
Vehicle	Collected reward	# of fulfilled orders	Traveling cost	Profit	Collected reward	# of fulfilled orders	Traveling cost	Profit
TC1	**152.3**	**25.3**	**23.1**	**129.2**	**159.0**	**27.3**	**18.6**	**140.4**
V1	168	33	22.8	145.2	241	44	28.4	212.6
V2	168	26	30.1	137.9	236	38	27.5	208.5
V3	121	17	16.5	104.5	-	-	-	-
TC2	**162.5**	**25.0**	**25.6**	**136.9**	**152.5**	**17.0**	**12.1**	**140.4**
V4	191	32	28.2	162.8	55	6	7.6	47.4
V5	134	18	23.0	111.0	250	28	16.6	233.4
TC3	**219.0**	**27.5**	**27.8**	**191.2**	**168.0**	**30.5**	**27.5**	**140.5**
V6	195	21	25.3	169.7	94	23	24.7	69.3
V7	243	34	30.2	212.8	242	38	30.3	211.8
TC4	**158.0**	**23.0**	**25.6**	**132.4**	**148.0**	**15.0**	**5.8**	**142.3**
V8	158	23	25.6	132.4	148	15	5.8	142.3
TC5	**153.7**	**19.7**	**32.1**	**121.5**	**161.0**	**23.7**	**20.7**	**140.3**
V9	170	25	28.4	141.6	250	33	33.3	216.7
V10	136	18	40.2	95.8	233	38	28.7	204.3
V11	155	16	27.7	127.3	-	-	-	-
TC6	**133.7**	**20.7**	**25.3**	**108.4**	**160.3**	**25.7**	**20.0**	**140.3**
V12	105	19	24.9	80.1	245	28	26.7	218.3
V13	167	26	20.7	146.3	236	49	33.4	202.6
V14	129	17	30.2	98.8	-	-	-	-
TC7	**207.0**	**27.5**	**26.2**	**180.8**	**168.5**	**26.0**	**27.2**	**141.3**
V15	236	26	31.4	204.6	93	12	23.3	69.7
V16	178	29	20.9	157.1	244	40	31.1	213.0
TC8	**116.0**	**16.0**	**25.8**	**90.2**	**158.3**	**22.0**	**18.1**	**140.3**
V17	120	18	26.9	93.1	242	28	24.7	217.3
V18	77	15	23.9	53.1	233	38	29.5	203.5
V19	151	15	26.7	124.3	-	-	-	-
TC9	**203.0**	**44.0**	**26.1**	**176.9**	**163.0**	**14.0**	**22.4**	**140.6**
V20	203	44	26.1	176.9	163	14	22.4	140.6
Total	3205.0	472.0	529.8	2675.2	3205.0	472.0	393.9	2811.1

(a)

(b)

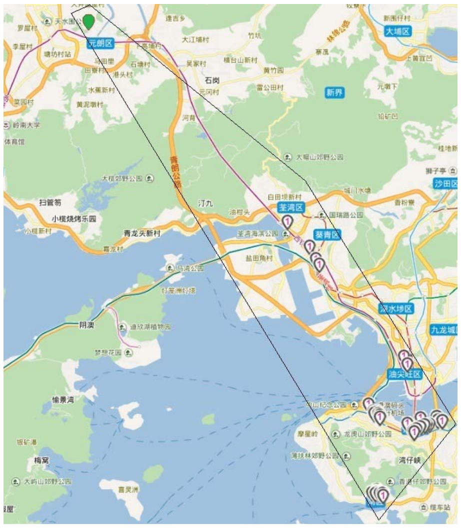
Fig. 10. Assignment of orders to vehicles by the scheduler. (a) New Territories. (b) Hong Kong Island.

of 26 non-dominated solutions. Two extreme solutions are $(313.6, 67.3)$ and $(393.9, 140.3)$, where the latter one is the most balanced solution. We have analyzed the structures of the most balanced solution and the manually generated solution. The comparison between these two solutions is presented in Table VIII, where TCx and Vy represent transport company x and vehicle y, respectively, and the row (marked in bold) associated with the transport company presents the average values over all its vehicles.

Table VIII shows that the balanced solution only uses 16 vehicles with the total traveling cost of 393.9. Roughly speaking, for most of the VRPs, fewer vehicles usually correspond to less total traveling cost. In the balanced solution found by MOLS+, the unit profit for each transport company lies in interval [140.3, 142.3], which is a small range and demonstrates that this solution treats all transport companies fairly. We further plot in Figure 11 the order assignment of several vehicles according to our solution. This figure reveals that the orders fulfilled by the same vehicle form a fan-shaped sector rather than a cluster. Such delivery pattern can reduce the traveling cost because "near-by" orders can be fulfilled in a convenient way when serving the orders with distant locations. In addition, it is much easier to ensure the fairness by adjusting the orders lied in adjacent fan-shaped sectors rather than those lied in adjacent clusters.

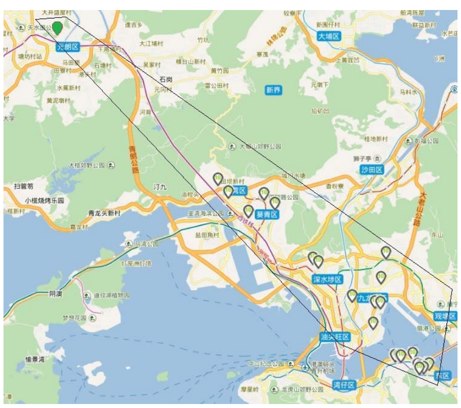
To sum up, our solution clearly dominates the manually generated solution in terms of both the total traveling cost and profit balancing. Our managerial suggestions to the food importer are as follows:

1) Improving the utilization of vehicles. Although our model does not consider the fixed costs of vehicles (e.g., the manpower cost), it is sufficient to conclude that cutting off extra vehicles can help reduce not only the traveling cost but also other fixed costs.
2) Optimizing the vehicle routes. Currently the food importer only concerns about the assignment of orders and lacks enough attention to vehicle routing. This case study shows that the traveling cost could be improved around 20%.



(a)

(b)



(c)

Fig. 11. Assignment of orders to vehicles. (a) Vehicle 1. (b) Vehicle 2. (c) Vehicle 10.

3) Considering fan-shaped delivery pattern. Although the clustering strategy may have some benefits for the management, it is not suitable for the optimization of both the traveling cost and profit balancing. The structure of our solution can be used for reference and guide the scheduling in practice.

VII. Conclusion

This paper introduces a new variant of the vehicle routing problem (VRPOPB) in which customer orders are outsourced to multiple transport companies. Two objectives, namely, minimizing the total traveling cost and balancing the profit among all transport companies, must be simultaneously considered. The first objective is important, since it relates to operational cost saving as well as pollution reduction, which has been emphasized by the governments of Hong Kong and other big cities. The second objective is also considerable for the importer in its long-run business. To balance the profits among transport companies, we chose maximizing the minimum unit profit rather than minimizing the difference between the maximum and minimum unit profits, because the latter way may lead to many distorted solutions.

In order to find an approximate Pareto-optimal set for the problem, we developed two multi-objective evolutionary algorithms, where the first one (called MOLS) is a standard multi-objective local search algorithm that uses four local search operators and the second one (called MOLS+) enhances the first one by incorporating the giant-tour representation, the recombination operator and the large neighborhood search procedure. The effectiveness of our algorithms are demonstrated using three experiments and one case study. In the first experiment, we adapted MOLS+ to solve the vehicle routing problem with route balance (VRPRB) and then compared its results with those reported in Jozefowiez *et al.* [13]. This comparison indicates that MOLS+ is capable of producing high-quality solutions for the VRPRB. The second and third experiments were used to compare the performance between MOLS and MOLS+ on the 57 instances derived from 7 VRPRB instances and 50 VRP instances. Moreover, the second experiment also studied the impacts of the giant-tour representation, the recombination operator and the large neighborhood search procedure. The computational results clearly show that these components can enhance the performance of the algorithm and MOLS+ is superior to MOLS. In the case study, we compared the plan produced by MOLS+ with the one made by a seasoned scheduler, and find that our solution dominates the manually generated one, i.e., our solution has smaller total traveling cost and the unit profits of all transport companies lie in a narrow interval. The results of this paper can serve as a baseline for future researchers working on related topics.

The VRPOPB is a very complex problem in transportation systems and our algorithms can produce near Pareto-optimal solutions. Future research directions may focus on designing more sophisticated heuristics for better solutions. Furthermore, our study can certainly be extended to deal with other practical situations; for example, multiple depots, time windows and multiple periods may be taken into account in the extended problems.

References

[1] S. Dehuri, A. K. Jagadev, and M. Panda, *Multi-objective Swarm Intelligence: Theoretical Advances and Applications*, vol. 592. Berlin, Germany: Springer-Verlag, 2015.

[2] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, “The orienteering problem: A survey,” *Eur. J. Oper. Res.*, vol. 209, no. 1, pp. 1–10, 2011.

[3] M. Keshtkaran, K. Ziarati, A. Bettinelli, and D. Vigo, “Enhanced exact solution methods for the team orienteering problem,” *Int. J. Prod. Res.*, vol. 54, no. 2, pp. 591–601, 2016.

[4] C. Archetti, M. G. Speranza, and D. Vigo, “Vehicle routing problems with profits,” in *Vehicle Routing: Problems, Methods, and Applications*, vol. 18. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2014, p. 273.

[5] Z. Zhang, O. Che, B. Cheang, A. Lim, and H. Qin, "A memetic algorithm for the multiperiod vehicle routing problem with profit," *Eur. J. Oper. Res.*, vol. 229, no. 3, pp. 573–584, 2013.

[6] H. Qin, W. Ming, Z. Zhang, Y. Xie, and A. Lim, "A tabu search algorithm for the multi-period inspector scheduling problem," *Comput. Oper. Res.*, vol. 59, pp. 78–93, Jul. 2015.

[7] I. Moon, J.-H. Lee, and J. Seong, "Vehicle routing problem with time windows considering overtime and outsourcing vehicles," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13202–13213, 2012.

[8] L. H. Lee, K. C. Tan, K. Ou, and Y. H. Chew, "Vehicle capacity planning system: A case study on vehicle routing problem with time windows," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 2, pp. 169–178, Mar. 2003.

[9] G. Zäpfel and M. Bögl, "Multi-period vehicle routing and crew scheduling with outsourcing options," *Int. J. Prod. Econ.*, vol. 113, no. 2, pp. 980–996, 2008.

[10] Y. Li, X. Wang, and T. M. Adams, "Ride service outsourcing for profit maximization," *Transp. Res. E, Logistics Transp. Rev.*, vol. 45, no. 1, pp. 138–148, 2009.

[11] T.-R. Lee and J.-H. Ueng, "A study of vehicle routing problems with load-balancing," *Int. J. Phys. Distrib. Logistics Manage.*, vol. 29, no. 10, pp. 646–657, Dec. 1999.

[12] G. Nikolakopoulou, S. Kortesis, A. Synefaki, and R. Kalfakakou, "Solving a vehicle routing problem by balancing the vehicles time utilization," *Eur. J. Oper. Res.*, vol. 152, no. 2, pp. 520–527, 2004.

[13] N. Jozefowiez, F. Semet, and E.-G. Talbi, "An evolutionary algorithm for the vehicle routing problem with route balancing," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 761–769, 2009.

[14] M. N. Kritikos and G. Ioannou, "The balanced cargo vehicle routing problem with time windows," *Int. J. Prod. Econ.*, vol. 123, no. 1, pp. 42–51, 2010.

[15] M. Huang, K. Smilowitz, and B. Balcik, "Models for relief routing: Equity, efficiency and efficacy," *Transp. Res. E, logistics Transp. Rev.*, vol. 48, no. 1, pp. 2–18, 2012.

[16] J. Pacheco, R. Caballero, M. Laguna, and J. Molina, "Bi-objective bus routing: an application to school buses in rural areas," *Transp. Sci.*, vol. 47, no. 3, pp. 397–411, 2013.

[17] B. Melián-Batista, A. De Santiago, F. AngelBello, and A. Alvarez, "A bi-objective vehicle routing problem with time windows: A real case in tenerife," *Appl. Soft Comput.*, vol. 17, pp. 140–152, Apr. 2014.

[18] N. Jozefowiez, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems," *Eur. J. Oper. Res.*, vol. 189, no. 2, pp. 293–309, Sep. 2008.

[19] Y. Zhou and J. Wang, "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows," *IEEE Syst. J.*, vol. 9, no. 3, pp. 1100–1113, Sep. 2015.

[20] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. P. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2015.

[21] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, no. 4, pp. 326–329, 1960.

[22] D. G. Cattrysse and L. N. Van Wassenhove, "A survey of algorithms for the generalized assignment problem," *Eur. J. Oper. Res.*, vol. 60, no. 3, pp. 260–272, Aug. 1992.

[23] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[25] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[26] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems," *Eur. J. Oper. Res.*, vol. 172, no. 3, pp. 855–885, 2006.

[27] M. Liu, C.-Y. Lee, Z. Zhang, and C. Chu, "Bi-objective optimization for the container terminal integrated planning," *Transp. Res. B, Methodol.*, vol. 93, pp. 720–749, Nov. 2016.

[28] F. Tricoire, "Multi-directional local search," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3089–3101, 2012.

[29] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. Hoboken, NJ, USA: Wiley, 2001.

[30] P. Toth and D. Vigo, *The Vehicle Routing Problem*. Philadelphia, PA, USA: SIAM, 2001.

[31] B. E. Gillett and L. Miller, "A heuristic algorithm for the vehicle-dispatch problem," *Oper. Res.*, vol. 22, no. 2, pp. 340–349, 1974.

[32] Z. Zhang, H. Qin, and A. Lim, "A memetic algorithm for the capacitated m-ring-star problem," *Appl. Intell.*, vol. 40, no. 2, pp. 305–321, 2014.

[33] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Syst.*, vol. 9, no. 3, pp. 193–212, 1995.

[34] P. Moscato and C. Cotta, "A gentle introduction to memetic algorithms," in *Handbook of Metaheuristics*. Boston, MA, USA: Springer, 2003, pp. 105–144.

[35] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Comput. Oper. Res.*, vol. 31, no. 12, pp. 1985–2002, 2004.

[36] C. Prins, N. Labadi, and M. Reghioui, "Tour splitting algorithms for vehicle routing problems," *Int. J. Prod. Res.*, vol. 47, no. 2, pp. 507–535, 2009.

[37] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Berlin, Germany: Springer, 2007.

[38] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Principles and Practice of Constraint Programming—CP*. Berlin, Germany: Springer, 1998, pp. 417–431.

[39] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transp. Sci.*, vol. 40, no. 4, pp. 455–472, 2006.

[40] N. Christofides and S. Eilon, "An algorithm for the vehicle-dispatching problem," *J. Oper. Res. Soc.*, vol. 20, no. 3, pp. 309–318, 1969.

[41] P. Augerat, J. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi, "Computational results with a branch-and-cut code for the capacitated vehicle routing problem," Rapport de recherche Imag., Tech. Rep., 1995.

[42] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Program. Evolvable Mach.*, vol. 6, no. 2, pp. 163–190, 2005.

[43] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 1998, pp. 292–301.

Zizhen Zhang received the B.S. and M.S. degrees from the Department of Computer Science, Sun Yat-sen University, China, in 2007 and 2009, respectively, and the Ph.D. degree from the Department of Management Sciences, City University of Hong Kong, in 2014. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University. His research interest includes computational intelligence and its applications in industrial engineering and transportation.

Hu Qin received the Ph.D. degree from the City University of Hong Kong, Kowloon, in 2011. He is currently a Professor with the School of Management, Huazhong University of Science and Technology. His current research interests include algorithms and artificial intelligence, including various topics in operations research, such as vehicle routing problem, freight allocation problem, container loading problems, and transportation problems.

Yanzhi Li received the bachelor's degree from the Computer Science Department, Tsinghua University, in 2002, and the Ph.D. degree from the Department of Industrial Engineering and Logistics Management, Hong Kong University of Science and Technology, in 2006. He is currently a Professor with the College of Business, City University of Hong Kong. His research is primarily on logistics, and operations and supply chain management, with a recent focus on data-driven decision making.